

TITLE OF THE INVENTION

Method and Apparatus for Using Non-Secure File Servers for
Secure Information Storage

5

CROSS REFERENCE TO RELATED APPLICATIONS

N/A

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR

10

DEVELOPMENT

N/A

BACKGROUND OF THE INVENTION

15 The present invention relates to techniques for
securely storing information on a non-secure file server and
distributing the securely stored information among clients
authorized to read and modify the information.

20 In communications networks, file servers are typically
employed to store files accessible over the network. With
the advancements in digital data storage and the
comparatively low cost of data storage, it is now
commonplace to have one or more file servers that have large
data storage capacities. Networks employed to interconnect
various clients with the file server are often not secure
25 and the file server itself is often not secure. Moreover,
there are certain applications in which it is desirable to
store data on file servers administered by an organization
independent of the clients that have a need to store the
data. For example, a company may desire, for purposes of
30 redundancy, to store their information on one file server
which is secure and located within the corporate environment
and another file server which is remote from and independent

-1-

ATTORNEY DOCKET NO. P4421
WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES, LLP
TEL. (617) 542-2290
FAX. (617) 451-0313

Express Mail No.

EL231112339US

of the corporate enterprise. This may be done to protect against the possibility of natural or other disasters which could destroy the information stored in the secure file server on the corporate environment. Additionally, it may
5 be more cost effective to outsource the storage function for large volume data storage. It would therefore be desirable in certain applications to be able to store data securely on a non-secure file server while being able to share the data among a number of clients that are authorized to have access
10 to the data.

BRIEF SUMMARY OF THE INVENTION

A method and apparatus for storing data securely on an non-secure file server is disclosed. The disclosed
15 technique prevents unauthorized users having access to the file server from obtaining intelligible information from the data stored on the file server and allows the data to be readily accessed by authorized clients or members of authorized groups. A first client desiring to store data on
20 the non-secure file server encrypts the data with a first encryption key having an associated first decryption key. The first decryption key is encrypted, in a preferred embodiment, with a second encryption key having an associated second decryption key. The encrypted data and
25 the encrypted first decryption key are forwarded from the first client to the file server for storage. The encrypted data is stored on the file server and the encrypted first decryption key is stored on the file server in an access control list associated with the encrypted data. In the
30 event other clients are to be provided access to the file, the first client or another client having access to the data encrypts the first decryption key with respective encryption

keys having associated decryption keys known to the respective clients and the additional encrypted first encryption keys are also stored on the non-secure file server in association with the encrypted data as entries in the access control list. In response to a request to access the encrypted data, the file server returns the encrypted data and at least the applicable encrypted first encryption key needed to decrypt the data. Alternatively, the file server returns the entire access control list. Groups of clients may be assigned a group encryption key and an associated group decryption key. The first decryption key may be encrypted using the group encryption key and the first decryption key encrypted with the group encryption key may be stored in the access control list. The first decryption key may then be decrypted by any group member or group server having access to the group decryption key. In response to a request to access the data, the file server may thus return either single entry in the access control list or the entire access control list. The requesting client or a group server decrypts the applicable encrypted first decryption key within the list to obtain the decryption key needed to decrypt the data.

Other features, aspects and advantages of the above-described techniques for securely storing data on a non-secure file server are described with particularity in the Detailed Description of the Invention which follows.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

The invention will be more fully understood by reference to the following Detailed Description of the Invention in conjunction with the Drawing of which:

Fig. 1 is a block diagram illustrating a plurality of clients coupled to a file server over a network and operative in a manner consistent with the present invention;

Fig. 2a is a block diagram of a client of the type depicted in Fig. 1;

Fig. 2b is a block diagram of a file server of the type depicted in Fig. 1;

Fig. 2c is a block diagram of a group server of the type depicted in Fig. 1;

Fig. 3a is a table illustrating an access control list including a single list entry;

Fig. 3b is a table illustrating an access control list including three entries within the access control list;

Fig. 3c is another example of an access control list that includes four entries within the access control list;

Fig. 3d is another example of an access control list that includes 5 entries within the access control list;

Fig. 4a is an illustrative message payload including a client identifier; an encrypted first decryption key and encrypted data;

Fig. 4b is an illustrative message payload including plural client identifiers; an associated encrypted first decryption key for each of the respective client identifiers and encrypted data;

Fig. 4c is an illustrative message payload including plural client identifiers, an associated encrypted first decryption key for each of the respective client identifiers, and encrypted data;

Fig. 4d is an illustrative message payload including plural client identifiers and a group identifier, an associated encrypted first decryption key for each of the respective client and group identifiers, and encrypted data;

Fig. 5 is a flow diagram illustrating a method employed at a client for encrypting data and an associated decryption key for storage on the file server of Fig. 1;

Fig. 6 is a flow diagram illustrating a method employed at a client for modifying an access control list stored on the file server of Fig. 1;

Fig. 7 is a flow diagram illustrating a method employed at the file server of Fig. 1 for responding to a request for data from a client;

Fig. 8 is a flow diagram illustrating a method employed at a client for decrypting encrypted data retrieved from the file server of Fig. 1;

Fig. 9a depicts an access control list including a single access control list entry;

Fig. 9b depicts an access control list including three access control list entries;

Fig. 9c depicts an access control list including four access control list entries;

Fig. 9d depicts an access control list including five access control list entries;

Fig. 10a is an illustrative message payload including a 10 check value and a first decryption key and data encrypted with a first encryption key;

Fig. 10b is an illustrative message payload including plural encrypted check values and first decryption keys and data encrypted with the first encryption key;

Fig. 10c is another illustrative message payload including plural encrypted check values and first decryption keys and data encrypted with the first encryption key; and

Fig. 10d is another illustrative message payload including plural encrypted check values and first decryption keys and data encrypted with the first encryption key.

DETAILED DESCRIPTION OF THE INVENTION

Consistent with the present invention a method and apparatus are disclosed for storing data securely on a non-secure file server while allowing the data to be accessed by a plurality of authorized clients and/or groups. The system is depicted generally in Fig. 1 in a simplified block diagram. The system includes a plurality of clients 12, identified as clients $C_a - C_n$ and a file server 14 communicatively coupled via a network 10. Additionally, certain embodiments of the system may include a group server 16 which is communicatively coupled to the network 10.

The network 10 may comprise a local area network, a wide area network, the Internet or any other network for communicatively coupling the respective clients 12, the file server 14 and the group server 16. The clients 12 are depicted generally in Fig. 2a and may comprise a computer or processing element, a personal digital assistant (PDA), an intelligent networked appliance, a controller or other device capable of storing and retrieving information to and from the file server 14. More specifically, the clients 12 typically include a processor 12a which is operative to execute programmed instructions out of an instruction memory 12b. The instructions executed in performing the functions herein described may comprise instructions stored within program code considered part of the operating systems 12e, instructions stored within program code considered part of an application 12f or instructions stored within program code allocated between the operating system 12e and the application 12f. The memory 12b may comprise random

access memory or a combination of random access memory and read only memory. The clients 12 include a network interface 12d for coupling the respective client to the network 10 and may optionally include secondary storage 5 12c.

The file server 14 is depicted generally in Fig. 2b and may comprise a storage subsystem in the form of an intelligent hard disk array or any other data storage subsystem suitable for accessing data in response to 10 requests issued to the file server 14 by clients communicably coupled to the file server 14 via the network 10. More specifically, the file server 14 typically includes a processor 14a which is operative to execute programmed instructions out of an instruction memory 14b. 15 The instructions executed in performing the file server functions herein described may comprise instructions stored within program code considered part of the file server operating systems 14e, instructions stored within program code considered part of a file server application 20 14f or instructions stored within program code allocated between the file server operating system 14e and the file server application 14f. The memory 14b may comprise random access memory or a combination of random access memory and read only memory. The file server 14 includes 25 a network interface 14d for coupling the respective client to the network 10 and includes secondary storage 14c for storing encrypted data forwarded from the clients 12 along with associated access control lists as discussed hereinafter in greater detail.

30 The group server 16 is generally depicted in Fig. 2c. The group server 16 is employed in specific embodiments which support the distribution of encrypted data among

groups having a plurality of clients or other groups as members. The group server 16 typically includes a processor 16a which is operative to execute programmed instructions out of an instruction memory 16b. The instructions executed in performing the group server functions herein described may comprise instructions stored within program code considered part of the group server operating systems 16e, instructions stored within program code considered part of a group server application 14f or instructions stored within program code allocated between the file server operating system 16e and the file server application 16f. The memory 16b may comprise random access memory or a combination of random access memory and read only memory. The group server 16 includes a network interface 16d for coupling the respective client to the network 10 and includes secondary storage 16c for storing encrypted data forwarded from the clients 12 along with associated access control lists as discussed hereinafter in greater detail.

A first system for securely storing data on the file server 14 and securely distributing the data among clients 12 authorized to obtain access to the data is described below with reference to Figs. 1, 3a-3d and 4a-4d. When a client, such as client C_a , desires to store data on a non-secure file server, such as the file server 14, the client C_a encrypts the data with a first encryption key K_{1e} having a corresponding first decryption key K_{1d} . For brevity, the first decryption key K_{1d} appears as the key K within the braces in the Figures and the first encryption key appears as the key K outside of the braces and as a subscript. In a preferred embodiment, the first encryption key and the first decryption key comprise a single symmetric key

although the first encryption key and the first decryption key may comprise public and private keys or a public key pair. For purposes of reference, the data to be stored on the file server 14 is referred to as data or file F.

5 After encrypting the data F, the client C_a encrypts the first decryption key K_{1d} with a second encryption key K_{2e} having a corresponding second decryption key K_{2d} known to the client C_a . In a preferred embodiment the second encryption key K_{2e} comprises the public key of a public key pair owned by the client C_a . Thus, the encrypted first decryption key can only be decrypted by the client 12 which owns or has access to the private key of the public key pair, which typically would be only client C_a . The encrypted data and the encrypted first decryption key K_{1d}

10 along with a client or group identifier are forwarded over the network 10 by the client 12 C_a for receipt by the file server 14. As illustrated in Fig. 3a, the file server 14 stores the received encrypted first decryption key along with the client identifier C_a in an access control list.

15 The access control list may be stored separately from the associated encrypted data as depicted in Fig. 3a or alternatively, stored in a header along with the encrypted data file as illustrated in Fig. 4a.

Since, in the present example, the first decryption

25 key has been encrypted with only the second encryption key having a second decryption key owned by client C_a only client C_a can presently obtain access to the encrypted data. The client C_a 12 may access the data stored on the file server 14 as follows. The client C_a issues a request

30 to the file server 14 to access the data. In the present embodiment, the file server 14 compares the client identifier C_a to the client identifiers stored within the

access control list as depicted in Fig. 3a. Since the client identifier of the client 12 issuing the request to the file server 14 matches the client identifier C_a in the access control list, the file server 14 retrieves the corresponding encrypted first decryption key (encrypted with the public key of client C_a) and returns the encrypted first decryption key along with the encrypted data to the client C_a . The client C_a , upon receipt of the encrypted first decryption key K_{1d} (which was encrypted with the client C_a 's public key), and the encrypted data, decrypts the encrypted first decryption key K_{1d} to obtain the unencrypted first decryption key and then decrypts the data F using the first decryption key. In this manner, the data F may be securely stored on the file server 14 and retrieved by the client 12 that initially stored the data. While in the above example, the file server 14 compared the client identifier of the client requesting the data to the client identifier within the access control list to identify the encrypted first decryption key to be returned, the file server 14 may return to the requesting client the entire access control list.

In many circumstances, it is desirable to be able to have one client 12 store data on the file server 14 while allowing other authorized clients 12 to access the data. The following example illustrates key distribution mechanisms which allow keys to be distributed so that encrypted data stored on a non-secure file server 14 may be accessed by a number of authorized clients 12 and group members.

Referring to Figs. 3b and 4b, and continuing with example depicted in Figs. 3a and 4a, assume that client C_a desires to permit clients C_b and C_c access to the encrypted

data F stored on the file server 14 by the client C_a . In this circumstance, the client C_a retrieves the access control list for the encrypted data F stored on the file server 14, obtains the first decryption key by decrypting the encrypted first decryption key and encrypts the first decryption key K_{1d} with each of the public keys of the public key pairs of clients 12 C_b and C_c . The client C_a then appends the additional encrypted first decryption keys to the access control list along with the client identifiers for the respective encrypted first decryption keys as depicted in Fig. 3b. The modified access control list is forwarded by the client C_a to the file server 14 and the modified access control list is stored on the file server 14 as indicated in Fig. 3b. Alternatively, the modified access control list is stored as a header in conjunction with the encrypted data F as depicted in Fig. 4b. In response to a request to access the data F from any one of the authorized clients C_a , C_b or C_c , the file server compares the client identifier of the requesting client and returns the applicable encrypted first decryption key along with the encrypted data F or alternatively, returns to the requesting client the entire access control list as it then exists along with the encrypted data F. If a single encrypted first decryption key is returned, the requesting client 12 decrypts the encrypted first decryption key with the private key of its public key pair and utilizes the decrypted first decryption key to decrypt the encrypted data F. If the entire access control list is returned to the requesting client, the requesting client compares its client identifier to the client identifiers within the access control list to select the applicable encrypted first

decryption key to be employed in the data decryption process. In the above-described manner, the client 12 that initially stored the encrypted data F has added to the access control list encrypted decryption key entries which allow other authorized clients to decrypt the stored data.

Continuing with the present example, and as depicted in Figs. 3c and 4c, clients 12 that are authorized to access the data F may authorize other clients 12 to access the data F. For example, assume that client C_c, which has been authorized to access the encrypted data F, desires to add client C_d to the access control list so that client C_d is authorized to access the encrypted data F. In this circumstance, as described above, client C_c retrieves the access control list associated with the data F. Client C_c then retrieves the unencrypted first decryption key by decrypting the encrypted first decryption key that had been encrypted utilizing C_c's public key. Client C_c then encrypts the first decryption key using client C_d's public key. After encrypting the first decryption key with client C_d's public key, the client identifier C_d and the encrypted first decryption key for client C_d is appended to the access control list and the modified access control list is forwarded to the file server 14. The modified access control list is stored on the file server 14 as depicted in Fig. 3c or alternatively, in conjunction with the encrypted data as depicted in Fig. 4c.

In response to a request to access the encrypted data F initiated by client C_d, the file server returns to the client C_d either the applicable encrypted first decryption key which was encrypted using client C_d's public key along with the encrypted data F or alternatively, the entire

current access control list along with the encrypted data F as discussed above. It should be noted that the access control list may be modified by appending the new entries to the list rather than forwarding the entire modified access control list to the file server 14 for storage.

Groups of clients 12 may also be authorized to obtain access to the encrypted data F stored on the file server 14. For example, referring to Fig. 1 assume that a group G_1 is composed of clients C_e and C_f . Member clients 12 belonging to the group G_1 may be provided access to the encrypted data F stored on the file server 14 as described below. The group G_1 is provided with a public key pair comprising a group public key and a group private key. In a first embodiment for servicing groups of clients, each of the member clients within the group G_1 is provided with the group private key so that each group member can decrypt information encrypted using the group public key. A client or a client member of a group previously authorized to obtain access to the encrypted data F may add the group G_1 to the access control list in the manner previously described. More specifically, continuing with the prior example, assume that client C_d desires to add group G_1 to the access control list. In this circumstance, client C_d retrieves either its own encrypted first decryption key, or alternatively, retrieves the then current access control list from the file server 14. The client C_d decrypts the encrypted first decryption key to obtain the unencrypted first decryption key and encrypts the first decryption key using the public key of the group G_1 . The first decryption key encrypted with the public key of group G_1 is then appended to the access control list and the modified access control list is forwarded to the file

server 14. Alternatively, the client C_d may forward to the client server 14 the encrypted first decryption key for the group G_1 to be appended to the access control list as depicted in Fig. 3d or stored in conjunction with the encrypted data F as illustrated in Fig. 4d. In response to a request to access the file from a client 12 which is a member of the group G_1 , the file server returns the applicable encrypted first decryption key for the group along with the encrypted data F or alternatively, returns the entire access control list along with the encrypted data F as discussed above. The requesting client which is a member of the group G_1 , can decrypt the encrypted first decryption key to obtain the unencrypted first decryption key using the group private key and then decrypt the encrypted data F using the unencrypted first decryption key.

While the provision of the group private key to the group members has certain advantages in terms of processing efficiency, this approach also has certain disadvantages. In particular, if a group member ceases to be a member of the group, the public key pair for the group would need to be modified to prevent the former group member from accessing the files based upon the clients former membership in the group. Additionally, access control list entries for the group would need to be re-encrypted such that the list would contain the first decryption key encrypted with the new group public key.

In another embodiment which is operative to service requests for files from members of a group, a group server 16 depicted in Fig. 1 is employed to decrypt an encrypted copy of the first encryption key. A client 12 or group server 16 encrypts the first decryption key with the

public key of the group server 16 and modifies the access control list or the header of the encrypted data stored on the file server 14 to include the group identifier and the encrypted first decryption key as depicted in Figs 3d or 4d. In response to a request from a client 12 group G_1 member for access to data stored in encrypted form on the file server 14, the file server 14, as described above, returns the encrypted data to the requesting client 12 group G_1 member along with either the encrypted first decryption key (encrypted with the group server public key) for the group G_1 or alternatively, returns to the requesting client 12 group member the entire access control list. The requesting client 12, upon receipt of the encrypted key or the access control list, forwards at least the encrypted first decryption key for the group G_1 (encrypted with the group server 16 public key) to the group server 16. The group server 16 then determines whether the client that forwarded the encrypted first decryption key for decryption is a member of the respective group. If the client that forwarded the request is not a member of the group, the group server 16 does not proceed with the decryption of the encrypted first decryption key. If the client that forwarded the request is a member of the group, the group server 16 decrypts the encrypted first decryption key with the group server private key to obtain the unencrypted first decryption key. The group server 16 then forwards the first decryption key to the group member via a secure channel. The secure channel may comprise a physically secure channel or alternatively may comprise an encrypted message forwarded to the respective client over a non-secure communications link. For example, the group server

16 may encrypt the first decryption key with the public
key of the respective client 12 or alternatively with a
symmetric key shared between the group server 16 and the
respective client 12. If the first decryption key is
5 encrypted with the public key of the client 12, only the
client 12 having the corresponding private key can decrypt
the encrypted first decryption key which was encrypted
with respective clients public key. Upon receipt and
decryption of the encrypted first decryption key, the
10 client that initiated the request for data can utilize the
unencrypted first decryption key to decrypt the encrypted
data retrieved from the file server 14. The utilization
of a group server 16 to perform decryption and forwarding
of the first decryption key to the client group member
15 that requested data from the file server 14 avoids the
need to assign a new public key pair in the event that a
client ceases to be a group member and additionally,
avoids the need to update the access control list should a
group member exit the group. It should also be noted that
20 a symmetric key may be employed between the group server
16 and a client group member to establish the secure
channel over the non-secure communications link.

The method employed at a client 12 to store data on
the non-secure file server 14 while using non-encrypted
25 client and group identifiers to identify encrypted first
decryption keys in the access control list is illustrated
in Fig. 5. Referring to Fig. 5, the client 12 (or group
member) first obtains the data to be stored on the non-
secure file server 14 as illustrated in step 20. The data
30 may comprise a file generated by the client, such as a
text file or any other file generated by the client 12, a
database, information obtained by the client 12 from

another client, or any other form of information or data that the client desires to store on the non-secure file server 14. The client 12 next encrypts the data to be stored on the file server 14 with a first encryption key having an associated first decryption key as illustrated in step 22. The client 12 also encrypts the first decryption key with a second encryption key having an associated second decryption key known to the respective client or group as depicted in step 24. A first key identifier is associated with the encrypted first decryption key as shown in step 26. In the event the client 12 is storing the data for its own decryption, the key identifier would correspond to the respective client 12 identifier; i.e. if the client C_1 is storing the encrypted data on the non-secure file server for its own retrieval, the client C_1 associates the key identifier C_1 in unencrypted form with the encrypted first decryption key encrypted with the client C_1 second encrypted key. The client C_1 may encrypt the first encryption key with its own public key and use its own client identifier C_1 as the key identifier. In the event the data is first to be stored in the file server 14 on behalf of a group G_1 , the group member or group server that is storing the data on the file server 14 encrypts the first decryption key with a second encryption key having an associated second decryption key which is either published to the group members or alternatively, retained by the group server 14. The client 12 or group server 16 associates the group key identifier G_1 with the encrypted first decryption key and forwards the key identifier, the encrypted first decryption key and the encrypted data to the file server 14 for storage as illustrated in step 28.

Fig. 6 illustrates the method for modifying the access control list that is employed at a client (or group server if applicable) authorized to obtain access to the encrypted data stored on the non-secure file server 14.

5 Assume that a first client 12 that is authorized to access the stored data desires to authorize a second client to access the stored data. As depicted in step 40, the first client 12 that is authorized to access the stored data obtains the encrypted first decryption key associated with

10 its key identifier. This may occur in a number of ways. If the first client 12 is in possession of the first decryption key the first client need not obtain the encrypted first decryption key from the file server 14. In the event the first client 12 does not have the

15 encrypted first decryption key in its possession, or as a default, the first client 12 may request that the encrypted first decryption key associated with the first client's key identifier be returned from the file server 14. Alternatively, the first client 12 may request that

20 the then existing access control list (which includes the encrypted first encryption key encrypted with the first clients second encryption key) be returned from the file server 14. The objective of this step is simply to obtain at the first client a copy of the unencrypted first

25 encryption key. If the first client 12 retrieves an encrypted copy of the first encryption key from the file server 14, the client 12 decrypts the respective encrypted first decryption key with its second decryption key to obtain the unencrypted first decryption key. As

30 previously discussed, the second encryption and decryption keys are preferably the respective public and private keys of the public key pair owned by the respective client 12.

Alternatively, symmetric keys may be employed which are shared among the clients 12 that are authorized to access the stored data and authorized to modify the access control list.

5 After obtaining the unencrypted first decryption key, the first client 12 that desires to modify the access control list obtains a second encryption key of a second client 12 to be provided access to the stored data. The second client 12 second encryption key has an associated
10 second client second decryption key. As discussed above, the second client 12 second encryption key and second decryption key are preferably public and private keys of a public key pair owned by the added client. The first decryption key is encrypted by the first client 12 using
15 the second encryption key of the second client as depicted in step 42. The second client key identifier is then associated with the second client's encrypted first decryption key as depicted in step 44. Thereafter, the second client key identifier and the second client's
20 encrypted first decryption key are appended to the access control list as illustrated in step 46. This information may be appended to the access control list in a number of ways. For example, the access control list may be retrieved by the first client and the first client may
25 append the second client key identifier and encrypted first decryption key information to the prior access control list to form the modified access control list. Alternatively, the new key identifier and encrypted first decryption key may be forwarded to the file server 14 and
30 such information may be added to the access control list by the file server 14 in response to a request issued by the first client. Finally, if the access control list is

stored as a header to the encrypted data stored on the file server 14, the header is modified to correspond to the modified access control list. Thus, any client that is authorized to access the data stored on the non-secure
5 file server 14 may modify the access control list to authorize other clients or groups to access the stored data.

Fig. 7 illustrates the operation of the file server 14 in response to a request for stored data from a client
10 12 authorized to access such data. As indicated in step 60, a client authorized to access data stored on the non-secure file server 14 issues a request to the file server 14 to retrieve the data. In response, the file server 14 retrieves the encrypted data and at least the encrypted
15 first decryption key associated with the requesting client 12 as shown in step 62. More specifically, the file server 14, upon receipt of the request for data, may retrieve the data along with the relevant encrypted first decryption key by retrieving from the access control list
20 the key associated with the client or group identifier that issued the request. By retrieving only the needed key with the encrypted data and forwarding only the needed key to the requesting client 12, network bandwidth is conserved. Alternatively, the file server 14, in response
25 to a request for data, may retrieve the entire access control list and return the entire access control list and the encrypted data to the client. In either case, the file server 14 forwards to the client 12 that issued the request for the data the encrypted data along with at
30 least the relevant encrypted first decryption key as depicted in step 64.

Prior to forwarding the encrypted data and the encrypted first decryption key to a requesting client the file server 14 may perform a test to authenticate the client and ascertain whether the requesting client is included on the access control list. If the client is not included on the access control list, the file server may decline to return to the requesting client the encrypted data and/or the encrypted first decryption keys.

Fig. 8 illustrates the operation of a client 12 that desires to retrieve data stored on the non-secure file server 14. As illustrated at step 80, the client 12 (e.g. client C_b or a group member such as C_e in group G₁) that desires to retrieve data from the file server 14 issues a request to the file server for the particular data. Typically, the request will be in the form of a request to access data having a known file name. As illustrated in step 82, in response to the request, the requesting client receives from the file server 14 at least one encrypted decryption key for decrypting the encrypted data along with the encrypted data. As discussed above with respect to Fig. 7, the client 12 that issued the request may receive a single encrypted first decryption key (if the file server parses the request to identify the needed key from the received access control list) or alternatively, the full access control list (in which case, the requesting client identifies the needed key). Additionally, as discussed above, if the requesting client is a member of a group, the encrypted key or access control list, as applicable, may be forwarded to the group server 16 for decryption of the relevant encrypted first decryption key and the unencrypted first decryption key securely communicated to the respective group member. The

requesting client 12 or group member thus obtains an unencrypted copy of the first decryption key as depicted in step 84 and the unencrypted first decryption key is utilized to decrypt the encrypted data as shown in step 86.

In the embodiments described above, the client and group identifiers stored in the access control list or data header on the file server 14 are stored in unencrypted form or, as referred to in the trade, "in the clear". Thus, the file server 14 and users having access to the file server can access the identity of the clients and groups that are authorized to access given data even if such users cannot access the encrypted data and encrypted key associated with the respective identifiers.

In certain environments it may be desirable to preclude the file server and unauthorized users from obtaining even client and group identifying information. To prevent access to client and group identifiers while still permitting secure storage of data on a non-secure file server, variations of the above-described techniques are employed. These embodiments are described with respect to access control list examples illustrated in Figs. 9a through 9d and message payload data illustrated in Figs. 10a through 10d.

By way of example, assume that client C_a desires to store data on the non-secure file server 14. The client C_a encrypts the data with a first encryption key having an associated first decryption key. As discussed above, the first encryption and decryption keys are preferably a symmetric key, but a public key pair may be employed. The client C_a appends the unencrypted first decryption key "K" to an unencrypted client identifier or unencrypted check

value "X" and encrypts the data stream {XK} with a second encryption key having an associated second decryption key known to client C_a. The second encryption key in a preferred embodiment is the public key of a public key pair owned by the client C_a. The value X may be the client identifier (or group identifier) for the client C_a storing the data or alternatively, a secret value or a value known to the client C_a. For purposes of illustration in the present example it will be assumed that the value X is the client identifier C_a and the client identifier occupies a predetermined number of bytes. The encrypted data stream along with the encrypted data is forwarded by the client C_a to the file server 14. The encrypted data stream {XA} is stored in an access control list as depicted in Fig. 9a and associated with the stored encrypted data. Alternatively, the encrypted data stream is stored as a header to the encrypted data as depicted in Fig. 10a.

In response to a request from the client C_a to read the data stored on the file server 14, the file server returns the access control list for the encrypted data along with the encrypted data. The client C_a then attempts to decrypt each encrypted data stream in the access control list with its second decryption key, which, in the present example, comprises the private key of client C_a. In the initial example depicted in Fig 9a the access control list includes a single entry. After decrypting the encrypted data stream with client C_a's second decryption key, client C_a strips off an initial predetermined number of bytes from the decrypted data stream and compares the initial byte value to the client identifier for the client C_a. If the values compare, the data which follows is the unencrypted first decryption

key. The first decryption key is then used to decrypt the encrypted data that was retrieved from the file server. If the values do not compare, the remaining encrypted data streams in the access control list are decrypted using the
5 respective second decryption key to determine which data stream includes the first decryption key that was encrypted with the respective client's second encryption key. Upon decrypting the proper data stream, the first encryption key is used to decrypt the encrypted data. It
10 is noted that the present embodiment in which group identifiers are not in the clear is more compute intensive since the encrypted data streams must be sequentially decrypted to locate the data stream that was initially encrypted using the respective client's second encryption
15 key. Additionally, as discussed hereinafter, the situation is more complex when the access control list includes an encrypted data stream which may only be decrypted by a group server.

By way of further example, assume that client C_a
20 desires to authorize clients C_b and C_c to access the encrypted file. This is accomplished by modifying the access control list. More particularly, the client C_a obtains the first decryption key. If the first decryption key has not been retained within the client C_a , client C_a
25 retrieves the access control list from the file server 14 and decrypts the entries in the access control list with client C_a 's second decryption key until the entry encrypted with client C_a 's second encryption key is located. After decrypting the data stream {XK} that includes the value X
30 which corresponds to the client's identifier, the corresponding first decryption key is used by the client C_a to generate new data stream entries for inclusion in a

modified access control list. In particular, referring to Fig. 9b, the first decryption key (K) is appended to the client identifier for client C_b (X) and the combined data stream is encrypted with the second encryption key for the client C_b. Additionally, the first decryption key (K) is appended to the client identifier for client C_c (X) and the combined data stream is encrypted with the second encryption key for the client C_c. The encrypted data streams are then added to the access control list as shown in Fig. 9b or included in the header of the encrypted data as illustrated in Fig. 10b.

In the event one of the clients 12 authorized to access the data stored in encrypted form on the file server 14 issues a request to the file server 14 to access the data, the file server returns the entire access control list along with the encrypted data. The requesting client 12 decrypts each successive encrypted data stream and tests the decrypted value X against its own client identifier (or check value) to identify the first decryption key that was encrypted with the respective client's second encryption key. The first decryption key is then used to decrypt the encrypted data.

Any client 12 that is authorized to access the data stored on the file server 14 can authorize another client 12 to access the stored data. For example, referring to Figs. 9c and 10c, any of clients C_a, C_b and C_c can modify the access control list to authorize client C_d to access the stored data as described above with respect to Figs. 9b and 10b.

Groups of clients may also be authorized to access data stored on the non-secure file server 14 without using un-encrypted group identifiers. Assume that client C_c

desires to authorize group G_1 to access the encrypted data stored on the file server 14. In such event, client C_c obtains the first decryption key as discussed above. Client C_c then appends the first decryption key to the group identifier G_1 to form a data stream, encrypts the data stream with a second encryption key which has a corresponding second decryption key owned by the group G_1 . In a preferred embodiment, the second encryption key comprises the public key of the group G_1 and the second decryption key comprises the private key of the group G_1 . The second decryption key for the group G_1 may be distributed among the members of the group or alternatively, retained by a group server such as group server 16. The encrypted data stream is added to the access control list as depicted in Fig. 9d or alternatively, included the header of the encrypted data as depicted in Fig. 10d. In the event the group G_1 members are provided with the group G_1 second decryption key, a group member retrieves the stored data as follows. Assume group G_1 includes clients C_e and C_f . In the event client C_f issues a request to the file server 14 for the stored data, the file server 14 returns the entire access control list (e.g. the access control list depicted in Fig. 10d) along with the encrypted data. Client C_f may not be authorized to access the file individually, but may be authorized to access the data as a member of the group G_1 . In such event, the client C_f attempts to decrypt the data streams within the retrieved access control list using its client second decryption key. In such event, none of the values X will match the client C_f identifier following decryption of the data streams. The client C_f may then attempt to decrypt the encrypted data streams within the

access control list using the group G_1 second encryption key. This effort will result in the identification of the group identifier G_1 within one of the decrypted data streams. Upon identifying the group identifier G_1 client
5 C_f uses the associated first decryption key to decrypt the encrypted data.

To avoid the problems discussed above with respect to the distribution of the group second decryption key, instead of distributing the group second decryption key to
10 the group members, the key may be retained by the group server 14. If the client C_f is unable to decrypt an encrypted data stream within the access control list using its own second decryption key (e.g. the client C_f private key), client C_f forwards the access control list to the
15 group server 14 which attempts to decrypt the encrypted data streams within the access control list using the group server second decryption key (e.g. the group G_1 private key). Upon recognizing the group G_1 identifier, the group server extracts the first decryption key from
20 the applicable data stream and forwards the first decryption key via a secure channel to the client C_f that issued the initial request for the data stored on the non-secure file server 14. The first decryption key may be forwarded over a secure physical channel or alternatively,
25 may be encrypted with an encryption key having an associated decryption key known to the client C_f . The client C_f may then utilize the first decryption key obtained from the group server 16 to decrypt the encrypted data.

30 While in the above example, client C_f attempted to decrypt the data streams within the retrieved access control list before forwarding the list to the group

server 16, it should be appreciated that the client C_f may forward the access control list initially to the group server 14 and not initiate decryption of data streams within the retrieved access control list until results are first received from the group server 16. Alternatively, the client C_f may initiate decryption of the data streams within the retrieved access control list with its own second decryption key while forwarding the access control list to the group server 16 for decryption of the data streams within the access control list in parallel. In this manner, the average time to obtain the first decryption key, and thus, to decrypt the encrypted data may be reduced.

The above described examples illustrate the use of encrypted client and group identifiers. As mentioned above, a password or known check value may be employed as the value X to which the first decryption key is appended. The known value will only be generated by a client decrypting the data stream with a second decryption key for which the respective data stream was encrypted using the corresponding second encryption key.

While in the above-described embodiments, the file server 14 transmits encrypted information to a client in response to a request issued by the respective client, it should be appreciated that the file server may transmit such information in the absence of a specific request from a particular client. For example, the file server 14 may periodically transmit encrypted information along with one or more of the access control list entries to one or more of the clients 12. Additionally, such transmission may occur on a non-periodic basis in response to specified events.

Those skilled in the art should readily appreciate that computer programs operative to perform the functions herein described can be delivered to a client, or file server or group server in many forms; including, but not limited to: (a) information permanently stored in a non-writable storage media (e.g. read-only memory devices within a computer such as ROM or CD-ROM disks readable by a computer I/O attachment); (b) information alterably stored on writable storage media (e.g. floppy disks, tapes, read/write optical media and hard drives); or (c) information conveyed to a computer through a communication media, for example, using baseband or broadband signaling techniques, such as over computer or telephone networks via a modem. In addition, it should be appreciated that the presently described methods may be implemented in software executing out of a memory on respective client, file or group server processors. Alternatively, the presently described functions may be embodied in whole or in part using hardware components such as Application Specific Integrated Circuits (ASICs), state machines, controllers or other hardware components or devices, or a combination of hardware components and software processes without departing from the inventive concepts herein described.

Those of ordinary skill in the art should further appreciate that variations to and modifications of the above-described methods and systems for granting access to a computer resource may be made without departing from the inventive concepts disclosed herein. Accordingly, the invention should be viewed as limited solely by the scope and spirit of the appended claims.